

[illegible]

```

SSSSSSSS  UU      UU  BBBB BBBB  MM      MM  IIIIII  TTTTTTTTTT
SSSSSSSS  UU      UU  BBBB BBBB  MM      MM  IIIIII  TTTTTTTTTT
SS      SS  UU      UU  BB      BB  MMMM  MMMM  II      TT
SS      SS  UU      UU  BB      BB  MMMM  MMMM  II      TT
SS      SS  UU      UU  BB      BB  MM  MM  MM  II      TT
SSSSSS  SS  UU      UU  BBBB BBBB  MM      MM  II      TT
SSSSSS  SS  UU      UU  BBBB BBBB  MM      MM  II      TT
SS      SS  UU      UU  BB      BB  MM      MM  II      TT
SS      SS  UU      UU  BB      BB  MM      MM  II      TT
SS      SS  UU      UU  BB      BB  MM      MM  II      TT
SSSSSS  UUUUUUUUUU  BBBB BBBB  MM      MM  IIIIII  TT
SSSSSS  UUUUUUUUUU  BBBB BBBB  MM      MM  IIIIII  TT

```

  

```

LL      IIIIII  SSSSSSSS
LL      IIIIII  SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLL  IIIIII  SSSSSSSS

```



```
1 0001 0 MODULE SUBMIT (%TITLE 'PRINT and SUBMIT Commands'
2 0002 0 MAIN = SUBMIT,
3 0003 0 IDENT = 'V04-000'
4 0004 0 ) =
5 0005 1 BEGIN
6 0006 1
7 0007 1 |*****
8 0008 1 |*
9 0009 1 |* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
10 0010 1 |* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
11 0011 1 |* ALL RIGHTS RESERVED.
12 0012 1 |*
13 0013 1 |* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
14 0014 1 |* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
15 0015 1 |* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
16 0016 1 |* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
17 0017 1 |* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
18 0018 1 |* TRANSFERRED.
19 0019 1 |*
20 0020 1 |* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
21 0021 1 |* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
22 0022 1 |* CORPORATION.
23 0023 1 |*
24 0024 1 |* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
25 0025 1 |* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
26 0026 1 |*
27 0027 1 |*
28 0028 1 |*****
29 0029 1 |
30 0030 1 |++
31 0031 1 |FACILITY:
32 0032 1 |PRINT and SUBMIT commands.
33 0033 1 |
34 0034 1 |ABSTRACT:
35 0035 1 |This module implements the PRINT and SUBMIT commands.
36 0036 1 |
37 0037 1 |ENVIRONMENT:
38 0038 1 |VAX/VMS user mode.
39 0039 1 |--
40 0040 1 |
41 0041 1 |AUTHOR: M. Jack, CREATION DATE: 30-Apr-1982
42 0042 1 |
43 0043 1 |MODIFIED BY:
44 0044 1 |
45 0045 1 |V03-009 RRB0021 Rowland R. Bradley 27-Aug-1984
46 0046 1 |Disallow the submission of command procedures and
47 0047 1 |and print jobs from sequential devices. Avoids the
48 0048 1 |problem described in Crash 201 (symbiont in RFAST state).
49 0049 1 |We do not support sharing sequential devices (mag tape).
50 0050 1 |Clean up the error logging associated with PRINT and
51 0051 1 |SUBMIT.
52 0052 1 |
53 0053 1 |V03-008 JAK0220 J A Krycka 21-Jul-1984
54 0054 1 |Make a $SNDJBC item list entry if /FEED is explicitly present.
55 0055 1 |Before, an item list entry was made only for /NOFEED because
56 0056 1 |/FEED was the default condition. Now the user can establish a
57 0057 1 |queue level default value for this attribute, so it is
```

SUBMIT  
V04-000

PRINT and SUBMIT Commands

K 10  
16-Sep-1984 00:28:51  
14-Sep-1984 12:09:55

VAX-11 Bliss-32 V4.0-742  
[CLIUTL.SRC]SUBMIT.B32;1

Page 2  
(1)

```

58 0058 1  necessary to be able to override the queue default.
59 0059 1
60 0060 1 V03-007 RRB0020 Rowland R. Bradley 18-Jun-1984
61 0061 1 Add support for common command qualifier processing
62 0062 1 This is a late V4 enhancement ask for by customers
63 0063 1 to allow dumb user file filtering.
64 0064 1 Add enhanced error message handling for /REMOTE
65 0065 1 and remove redundant error message when no files
66 0066 1 in the job. Add REMOINSF error code.
67 0067 1
68 0068 1 V03-006 BLS0267 Benn Schreiber 31-Jan-1984
69 0069 1 Use new LIB$FIND_FILE argument.
70 0070 1
71 0071 1 V03-005 MLJ0115 Martin L. Jack, 29-Jul-1983 13:40
72 0072 1 Changes for job controller baselevel.
73 0073 1
74 0074 1 V03-004 MLJ0113 Martin L. Jack, 26-May-1983 17:00
75 0075 1 Changes for job controller baselevel.
76 0076 1
77 0077 1 V03-003 MLJ0112 Martin L. Jack, 29-Apr-1983 0:17
78 0078 1 Changes for job controller baselevel.
79 0079 1
80 0080 1 V03-002 MLJ0109 Martin L. Jack, 13-Apr-1983 14:44
81 0081 1 Disallow node name without /REMOTE.
82 0082 1
83 0083 1 V03-001 MLJ0106 Martin L. Jack, 1-Mar-1983 5:03
84 0084 1 Changes for job controller baselevel.
85 0085 1
86 0086 1 **

```



```
88 0087 1 LIBRARY 'SYSS$LIBRARY:STARLET';
89 0088 1 REQUIRE 'SRC$:JBCPRSDEF';
90 0198 1
91 0199 1
92 0200 1 LITERAL
93 0201 1 TRUE= 1;
94 0202 1 FALSE= 0;
95 0203 1
96 0204 1
97 0205 1 STRUCTURE
98 0206 1 BBLOCK[O,P,S,E;N]=
99 0207 1 [N]
100 0208 1 (BBLOCK + 0)<P,S,E>;
101 0209 1
102 0210 1
103 0211 1 PSECT
104 0212 1 CODE= CODE,
105 0213 1 PLIT= CODE,
106 0214 1 OWN= DATA,
107 0215 1 GLOBAL= DATA;
108 0216 1
109 0217 1
110 0218 1 FORWARD ROUTINE
111 0219 1 SUBMIT,
112 0220 1 GET_COMM_QUAL_CONTEXT: NOVALUE,
113 0221 1 EXIT_HANDLER: NOVALUE,
114 0222 1 HANDLER;
115 0223 1
116 0224 1
117 0225 1 EXTERNAL ROUTINE
118 0226 1 CLISGET_VALUE: ADDRESSING_MODE(GENERAL),
119 0227 1 ! Get a parameter or qualifier value
120 0228 1 CLISPRESENT: ADDRESSING_MODE(GENERAL),
121 0229 1 ! Determine if entity is present
122 0230 1 LIB$FIND_FILE: ADDRESSING_MODE(GENERAL),
123 0231 1 ! Scan for wildcard files
124 0232 1 LIB$SCOPY_DXDX: ADDRESSING_MODE(GENERAL),
125 0233 1 ! Copy string
126 0234 1 LIB$QUAL_FILE_MATCH: ADDRESSING_MODE(GENERAL),
127 0235 1 ! Match common qualifiers
128 0236 1 LIB$QUAL_FILE_PARSE: ADDRESSING_MODE(GENERAL);
129 0237 1 ! Setup common qualifiers
130 0238 1
131 0239 1 EXTERNAL LITERAL
132 0240 1 ! Common Qualifier return status values.
133 0241 1
134 0242 1 LIB$QUIPRO, ! Quit Processing
135 0243 1 LIB$NEGANS, ! Negative Answer for this file
136 0244 1 LIB$FILFAIMAT, ! File failed to meet criteria
137 0245 1
138 0246 1 ! Parallel PRINT and SUBMIT messages.
139 0247 1
140 0248 1 PRINTS_FACILITY, SUBMITS_FACILITY,
141 0249 1 PRINTS_ADDFIL, SUBMITS_ADDFIL,
142 0250 1 PRINTS_CREJOB, SUBMITS_CREJOB,
143 0251 1 PRINTS_INVFILSPE, SUBMITS_INVFILSPE,
144 0252 1 PRINTS_REMOINSF, SUBMITS_REMOINSF,
```

SUBMIT  
V04-000

PRINT and SUBMIT Commands

M 10  
16-Sep-1984 00:28:51  
14-Sep-1984 12:09:55

VAX-11 Bliss-32 V4.0-742  
[CLIUTL.SRC]SUBMIT.B32;1

Page 4  
(2)

```
: 145      0253 1
: 146      0254 1      ! SUBMIT only messages.
: 147      0255 1
: 148      0256 1      SUBMIT$_INVLOGFIL;
: 149      0257 1
: 150      0258 1 GLOBAL
: 151      0259 1      QUAL_CONTEXT;      ! Context for Common Qualifiers
: 152      0260 1
: 153      0261 1 OWN
: 154      0262 1      EXIT_STATUS:      BBLOCK[4],
: 155      0263 1      FLAGS:      BBLOCK[4];      ! Most severe error status
: 156      0264 1      ! Flag bits
: 157      0265 1
: 158      0266 1 MACRO
: 159      0267 1      V_PRINT=      0,0,1,0 %;      ! PRINT command
: 160      0268 1      V_IDENTIFY=      0,1,1,0 %;      ! /IDENTIFY
: 161      0269 1      V_REMOTE=      0,2,1,0 %;      ! /REMOTE
: 162      0270 1      V_NO_LOG_FILE=      0,3,1,0 %;      ! /NOLOG_FILE
```



```
164 0271 1 ROUTINE SUBMIT=
165 0272 1
166 0273 1 ++
167 0274 1
168 0275 1 FUNCTIONAL DESCRIPTION:
169 0276 1 This routine is the main entry point for the PRINT and SUBMIT commands.
170 0277 1
171 0278 1 INPUT PARAMETERS:
172 0279 1 Standard VMS activation parameters (not used).
173 0280 1
174 0281 1 IMPLICIT INPUTS:
175 0282 1 NONE
176 0283 1
177 0284 1 OUTPUT PARAMETERS:
178 0285 1 NONE
179 0286 1
180 0287 1 IMPLICIT OUTPUTS:
181 0288 1 NONE
182 0289 1
183 0290 1 ROUTINE VALUE:
184 0291 1 Completion status.
185 0292 1
186 0293 1 SIDE EFFECTS:
187 0294 1 NONE
188 0295 1
189 0296 1 --
190 0297 1
191 0298 2 BEGIN
192 0299 2 LOCAL
193 0300 2 CNT: INITIAL(0), ! Number of files in job
194 0301 2 FIND_CONTEXT: REF BBLOCK, ! Context for LIB$FIND FILE
195 0302 2 FIND_RESULT: BBLOCK[DSC$C-S-BLN], ! Result of LIB$FIND FILE
196 0303 2 FIND_DEFAULT: BBLOCK[DSC$C-S-BLN], ! Default file for LIB$FIND FILE
197 0304 2 VALUE_DESC: BBLOCK[DSC$C-S-BLN], ! Qualifier value descriptor
198 0305 2 INPUT_DESC: BBLOCK[DSC$C-S-BLN], ! Input file specification
199 0306 2 LOG_FILE_DESC: BBLOCK[DSC$C-S-BLN], ! Log file descriptor
200 0307 2 JSM_BUFFER: VECTOR[132,BYTE], ! Job status message
201 0308 2 JSM_DESC: VECTOR[2], ! Descriptor for status message
202 0309 2 ITEM_BUFFER: BBLOCK[1024], ! $$NDJBC item buffer
203 0310 2 DATA_BUFFER: BBLOCK[1024], ! $$NDJBC data buffer
204 0311 2 EXIT_BLOCK: VECTOR[5], ! Exit control block
205 0312 2 MACRO
206 0313 2 SD_[A] = BIND %NAME('D_', A) = $DESCRIPTOR(A) %;
207 0314 2 SD_(
208 0315 2 '$VERB',
209 0316 2 'P1',
210 0317 2 'AFTER',
211 0318 2 'BURST',
212 0319 2 'CHARACTERISTICS',
213 0320 2 'CLI',
214 0321 2 'COPIES',
215 0322 2 'CPU TIME',
216 0323 2 'DELETE',
217 0324 2 'DEVICE',
218 0325 2 'FEED',
219 0326 2 'FLAG',
220 0327 2 'FORM',
```

```
.. 221 P 0328 2 'HEADER',
222 P 0329 2 'HOLD',
223 P 0330 2 'IDENTIFY',
224 P 0331 2 'JOB_COUNT',
225 P 0332 2 'KEEP',
226 P 0333 2 'LOG_FILE',
227 P 0334 2 'LOWERCASE',
228 P 0335 2 'NAME',
229 P 0336 2 'NOTE',
230 P 0337 2 'NOTIFY',
231 P 0338 2 'OPERATOR',
232 P 0339 2 'PAGES',
233 P 0340 2 'PARAMETERS',
234 P 0341 2 'PASSALL',
235 P 0342 2 'PRINTER',
236 P 0343 2 'PRIORITY',
237 P 0344 2 'QUEUE',
238 P 0345 2 'REMOTE',
239 P 0346 2 'RESTART',
240 P 0347 2 'SETUP',
241 P 0348 2 'SPACE',
242 P 0349 2 'TRAILER',
243 P 0350 2 'USER',
244 P 0351 2 'WSDEFAULT',
245 P 0352 2 'WSEXTENT',
246 P 0353 2 'WSQUOTA');
247 0354 2 BIND
248 0355 2 D_BURST_ALL= $DESCRIPTOR('BURST.ALL'),
249 0356 2 D_BURST_ONE= $DESCRIPTOR('BURST.ONE'),
250 0357 2 D_FLAG_ALL= $DESCRIPTOR('FLAG.ALL'),
251 0358 2 D_FLAG_ONE= $DESCRIPTOR('FLAG.ONE'),
252 0359 2 D_TRAILER_ALL= $DESCRIPTOR('TRAILER.ALL'),
253 0360 2 D_TRAILER_ONE= $DESCRIPTOR('TRAILER.ONE');
254 0361 2 BUILTIN
255 0362 2 FP;
256 0363 2
257 0364 2 PARSE_GLOBAL_REGISTERS;
258 0365 2
259 0366 2
260 0367 2 ! Initialize.
261 0368 2 !
262 0369 2 EXIT_STATUS = SS$NORMAL;
263 0370 2 .FP = HANDLER;
264 0371 2
265 0372 2 INPUT_DESC[DSC$B_CLASS] = DSC$K_CLASS_D;
266 0373 2 INPUT_DESC[DSC$B_DTYPE] = DSC$K_DTYPE_T;
267 0374 2 INPUT_DESC[DSC$W_LENGTH] = 0;
268 0375 2 INPUT_DESC[DSC$A_POINTER] = 0;
269 0376 2
270 0377 2 VALUE_DESC[DSC$B_CLASS] = DSC$K_CLASS_D;
271 0378 2 VALUE_DESC[DSC$B_DTYPE] = DSC$K_DTYPE_T;
272 0379 2 VALUE_DESC[DSC$W_LENGTH] = 0;
273 0380 2 VALUE_DESC[DSC$A_POINTER] = 0;
274 0381 2
275 0382 2 LOG_FILE_DESC[DSC$B_CLASS] = DSC$K_CLASS_D;
276 0383 2 LOG_FILE_DESC[DSC$B_DTYPE] = DSC$K_DTYPE_T;
277 0384 2 LOG_FILE_DESC[DSC$W_LENGTH] = 0;
```



```
278 0385 2 LOG_FILE_DESC[DSC$A_POINTER] = 0;
279 0386 2
280 0387 2 FIND_RESULT[DSC$B_CLASS] = DSC$K_CLASS_D;
281 0388 2 FIND_RESULT[DSC$B_DTYPE] = DSC$K_DTYPE_T;
282 0389 2 FIND_RESULT[DSC$W_LENGTH] = 0;
283 0390 2 FIND_RESULT[DSC$A_POINTER] = 0;
284 0391 2
285 0392 2 FIND_CONTEXT = 0;
286 0393 2
287 0394 2 FIND_DEFAULT[DSC$B_CLASS] = DSC$K_CLASS_S;
288 0395 2 FIND_DEFAULT[DSC$B_DTYPE] = DSC$K_DTYPE_T;
289 0396 2 FIND_DEFAULT[DSC$W_LENGTH] = %CHARCOUNT('COM');
290 0397 2 FIND_DEFAULT[DSC$A_POINTER] = UPLIT BYTE('COM');
291 0398 2
292 0399 2 FLAGS = 0;
293 0400 2
294 0401 2
295 0402 2 ! Initialize for command parsing utilities.
296 0403 2
297 P 0404 2 PARSE_GLOBAL_INIT(
298 P 0405 2     ICURSOR= ITEM_BUFFER,
299 P 0406 2     DCURSOR= DATA_BUFFER,
300 P 0407 2     MESSAGE= SUBMIT$ FACILITY*16 + SHR$_INVQUAVAL + STS$K_SEVERE,
301 0408 2     VALUE_DESC= VALUE_DESC);
302 0409 2
303 0410 2
304 0411 2 ! Get the command verb. If it is PRINT, rather than SUBMIT, change some
305 0412 2 defaults.
306 0413 2
307 0414 2 CLISGET VALUE(D $VERB, VALUE_DESC);
308 0415 2 IF CH$RCHAR(.VALUE_DESC[DSC$A_POINTER]) EQL %C'P'
309 0416 2 THEN
310 0417 2 BEGIN
311 0418 2     FLAGS[V_PRINT] = TRUE;
312 0419 2     FIND_DEFAULT[DSC$A_POINTER] = UPLIT BYTE('LIS');
313 0420 2 END;
314 0421 2
315 0422 2 ! Get the common qualifiers context for the PRINT command
316 0423 2
317 0424 2 GET_COMM_QUAL_CONTEXT();
318 0425 2
319 0426 2
320 0427 2 ! Get the command qualifiers.
321 0428 2
322 0429 2 FLAGS[V_IDENTIFY] = CLISPRESENT(D_IDENTIFY);
323 0430 2 FLAGS[V_REMOTE] = CLISPRESENT(D_REMOTE);
324 0431 2
325 0432 2
326 0433 2 ! Get the /AFTER qualifier.
327 0434 2
328 0435 2 PARSE_CALL(AFTER, D_AFTER);
329 0436 2
330 0437 2
331 0438 2 ! Get the /CHARACTERISTICS qualifier.
332 0439 2
333 0440 2 PARSE_CALL(CHARACTERISTICS, D_CHARACTERISTICS);
334 0441 2
```

```
335      0442 2
336      0443 2 ! Get the /HOLD qualifier.
337      0444 2
338      P 0445 2 PARSE_CALL(IF_TRUE, D_HOLD,
339      0446 2     SJC$_HOLD);
340      0447 2
341      0448 2
342      0449 2 ! Get the /NOTIFY qualifier.
343      0450 2
344      P 0451 2 PARSE_CALL(IF_TRUE, D_NOTIFY,
345      0452 2     SJC$_NOTIFY);
346      0453 2
347      0454 2
348      0455 2 ! Get the /PARAMETERS qualifier.
349      0456 2
350      0457 2 PARSE_CALL(PARAMETERS, D_PARAMETERS);
351      0458 2
352      0459 2
353      0460 2 ! Get the /PRIORITY qualifier.
354      0461 2
355      0462 2 PARSE_CALL(PRIORITY, D_PRIORITY);
356      0463 2
357      0464 2
358      0465 2 ! Get the /RESTART qualifier.
359      0466 2
360      P 0467 2 PARSE_CALL(IF_TRUE FALSE, D_RESTART,
361      0468 2     SJC$_RESTART, SJC$_NO_RESTART);
362      0469 2
363      0470 2
364      0471 2 ! Get the /USER qualifier.
365      0472 2
366      0473 2 PARSE_CALL(USER, D_USER);
367      0474 2
368      0475 2
369      0476 2 IF .FLAGS[V_PRINT]
370      0477 2 THEN
371      0478 2     BEGIN
372      0479 2
373      0480 2     ! Get the command-level /BURST qualifier.
374      0481 2
375      0482 2     PARSE_CALL(IF_TRUE, D_BURST_ALL, SJC$_FILE_BURST);
376      0483 2     PARSE_CALL(IF_TRUE, D_BURST_ONE, SJC$_FILE_BURST_ONE);
377      0484 2     PARSE_CALL(IF_FALSE, D_BURST, SJC$_NO_FILE_BURST);
378      0485 2
379      0486 2
380      0487 2     ! Get the /DEVICE qualifier.
381      0488 2
382      0489 2     IF CLISPRESNT(D_DEVICE) AND NOT CLISPRESNT(D_QUEUE)
383      0490 2     THEN
384      0491 2         PARSE_CALL(QUEUE, D_DEVICE, SJC$_QUEUE, $DESCRIPTOR('SYSS$PRINT'));
385      0492 2
386      0493 2
387      0494 2     ! Get the command-level /FLAG qualifier.
388      0495 2
389      0496 2     PARSE_CALL(IF_TRUE, D_FLAG_ALL, SJC$_FILE_FLAG);
390      0497 2     PARSE_CALL(IF_TRUE, D_FLAG_ONE, SJC$_FILE_FLAG_ONE);
391      0498 2     PARSE_CALL(IF_FALSE, D_FLAG, SJC$_NO_FILE_FLAG);
```



```
392 0499
393 0500
394 0501      ! Get the /FORM qualifier.
395 0502      !
396 0503      PARSE_CALL(FORM, D_FORM);
397 0504
398 0505
399 0506      ! Get the /JOB_COUNT qualifier.
400 0507      !
401 P 0508      PARSE_CALL(COPIES, D_JOB_COUNT,
402 0509      SJCS_JOB_COPIES);
403 0510
404 0511
405 0512      ! Get the /LOWERCASE qualifier.
406 0513      !
407 P 0514      PARSE_CALL(IF TRUE, D_LOWERCASE,
408 0515      SJCS_LOWERCASE);
409 0516
410 0517
411 0518      ! Get the /NAME qualifier.
412 0519      !
413 0520      PARSE_CALL(NAME, D_NAME);
414 0521
415 0522
416 0523      ! Get the /NOTE qualifier.
417 0524      !
418 P 0525      PARSE_CALL(STRING, D_NOTE,
419 0526      SJCS_NOTE, SJCS_NO_NOTE);
420 0527
421 0528
422 0529      ! Get the /OPERATOR qualifier.
423 0530      !
424 P 0531      PARSE_CALL(STRING, D_OPERATOR,
425 0532      SJCS_OPERATOR_REQUEST, SJCS_NO_OPERATOR_REQUEST);
426 0533
427 0534
428 0535      ! Get the /QUEUE qualifier.
429 0536      !
430 0537      IF NOT CLISPRESNT(D_DEVICE)
431 0538      THEN
432 0539          PARSE_CALL(QUEUE, D_QUEUE, SJCS_QUEUE, $DESCRIPTOR('SYSS$PRINT'));
433 0540
434 0541
435 0542      ! Get the command-level /TRAILER qualifier.
436 0543      !
437 0544      PARSE_CALL(IF TRUE, D_TRAILER_ALL, SJCS_FILE_TRAILER);
438 0545      PARSE_CALL(IF TRUE, D_TRAILER_ONE, SJCS_FILE_TRAILER ONE);
439 0546      PARSE_CALL(IF FALSE, D_TRAILER, SJCS_NO_FILE_TRAILER);
440 0547      END
441 0548      ELSE
442 0549      BEGIN
443 0550
444 0551      ! Get the /CLI qualifier.
445 0552      !
446 P 0553      PARSE_CALL(FILENAME, D_CLI,
447 0554      SJCS_CLI, SJCS_NO_CLI);
448 0555
```

```

449      0556      ! Get the /CPUTIME qualifier.
450      0557      !
451      0558      !
452      P 0559      PARSE_CALL(CPUTIME, D_CPUTIME,
453      0560      SJCS_CPU_LIMIT, SJCS_NO_CPU_LIMIT);
454      0561
455      0562
456      0563      ! Get the /KEEP qualifier.
457      0564      !
458      P 0565      PARSE_CALL(IF TRUE, D_KEEP,
459      0566      SJCS_NO_LOG_DELETE);
460      0567
461      0568
462      0569      ! Get the /LOG_FILE qualifier.
463      0570      !
464      0571      Q VALUE_DESC = LOG_FILE_DESC;
465      0572      FLAGS[V_NO_LOG_FILE] = PARSE_CALL_VALUE(LOG_FILE, D_LOG_FILE);
466      0573
467      0574
468      0575      ! Get the /NAME qualifier.
469      0576      !
470      0577      Q VALUE_DESC = VALUE_DESC;
471      0578      PARSE_CALL(NAME, D_NAME);
472      0579
473      0580
474      0581      ! Get the log file specification.
475      0582      !
476      0583      IF NOT .FLAGS[V_NO_LOG_FILE]
477      0584      THEN
478      P 0585      PARSE_CALL(NAME_AND_LOG_FILE,
479      P 0586      VALUE_DESC, LOG_FILE_DESC,
480      0587      $DESCRIPTOR('SYS$LOGIN:.LOG'), SUBMIT$_INVLOGFIL);
481      0588
482      0589
483      0590      ! Get the /PRINTER qualifier.
484      0591      !
485      0592      PARSE_CALL(PRINTER, D_PRINTER);
486      0593
487      0594
488      0595      ! Get the /QUEUE qualifier.
489      0596      !
490      0597      PARSE_CALL(QUEUE, D_QUEUE, SJCS_QUEUE, $DESCRIPTOR('SYS$BATCH'));
491      0598
492      0599
493      0600      ! Get the /WSDEFAULT qualifier.
494      0601      !
495      P 0602      PARSE_CALL(WORKING_SET, D_WSDEFAULT,
496      0603      SJCS_WSDEFAULT, SJCS_NO_WSDEFAULT);
497      0604
498      0605
499      0606      ! Get the /WSEXTENT qualifier.
500      0607      !
501      P 0608      PARSE_CALL(WORKING_SET, D_WSEXTENT,
502      0609      SJCS_WSEXTENT, SJCS_NO_WSEXTENT);
503      0610
504      0611
505      0612      ! Get the /WSQUOTA qualifier.

```



```
506      0613      !
507      P 0614      ! PARSE CALL(WORKING SET, D_WSQUOTA,
508      0615      !   SJCS_WSQUOTA, SJCS_NO_WSQUOTA);
509      0616      !
510      0617      !
511      0618      !
512      0619      ! Create the job.
513      0620      !
514      0621      ! IF NOT .FLAGS[V_REMOTE]
515      0622      ! THEN
516      0623      !   BEGIN
517      0624      !     LOCAL
518      0625      !       IOSB:          VECTOR[4,WORD], ! I/O status block
519      0626      !       STATUS:        ! Status return
520      0627      !
521      0628      !
522      0629      !   ! Finish the list and send the create job request.
523      0630      !
524      0631      !   Q_ICURSORE[0,0,32,0] = 0;
525      0632      !   STATUS = $SENDJBCW(
526      0633      !     FUNC=SJCS_CREATE_JOB,
527      0634      !     ITMLST=ITEM_BUFFER,
528      0635      !     IOSB=IOSB);
529      0636      !   IF .STATUS THEN STATUS = .IOSB;
530      0637      !   IF NOT .STATUS
531      0638      !   THEN
532      0639      !     SIGNAL_STOP(
533      0640      !       SUBMIT$_CREJOB, 0, .STATUS);
534      0641      !
535      0642      !
536      0643      !   ! Establish an exit handler to delete the open job on an abnormal exit.
537      0644      !
538      0645      !   EXIT_BLOCK[0] = 0;
539      0646      !   EXIT_BLOCK[1] = EXIT_HANDLER;
540      0647      !   EXIT_BLOCK[2] = 1;
541      0648      !   EXIT_BLOCK[3] = EXIT_BLOCK[4];
542      0649      !   EXIT_BLOCK[4] = 0;
543      0650      !   $DCLEXH(DESBLK=EXIT_BLOCK);
544      0651      !   END;
545      0652      !
546      0653      !
547      0654      !   ! Loop for all elements of the parameter list.
548      0655      !
549      0656      !   WHILE CLISGET_VALUE(D_P1, INPUT_DESC) DO
550      0657      !     BEGIN
551      0658      !       LOCAL
552      0659      !         STATUS_3,
553      0660      !         STATUS_2;
554      0661      !
555      0662      !       ! Set up the message buffer for the ADDFIL request.
556      0663      !
557      0664      !       Q_ICURSOR = ITEM_BUFFER;
558      0665      !       Q_DCURSOR = DATA_BUFFER;
559      0666      !
560      0667      !
561      0668      !       ! Get the /DELETE qualifier.
562      0669      !
```

```
563 P 0670 3 PARSE CALL(IF TRUE, D_DELETE,  
564 0671 3 SJCS_DELETE_FILE);  
565 0672 3  
566 0673 3  
567 0674 3 IF .FLAGS[V_PRINT]  
568 0675 3 THEN  
569 0676 4 BEGIN  
570 0677 4  
571 0678 4 ! Get the local-level /BURST qualifier.  
572 0679 4  
573 P 0680 4 PARSE CALL(LOCAL TRUE FALSE, D_BURST,  
574 0681 4 SJCS_FILE_BURST, SJCS_NO_FILE_BURST);  
575 0682 4  
576 0683 4  
577 0684 4 ! Get the /COPIES qualifier.  
578 0685 4  
579 P 0686 4 PARSE CALL(COPIES, D_COPIES,  
580 0687 4 SJCS_FILE_COPIES);  
581 0688 4  
582 0689 4  
583 0690 4 ! Get the /FEED qualifier.  
584 0691 4  
585 P 0692 4 PARSE CALL(IF TRUE FALSE, D_FEED,  
586 0693 4 SJCS_PAGINATE, SJCS_NO_PAGINATE);  
587 0694 4  
588 0695 4  
589 0696 4 ! Get the local-level /FLAG qualifier.  
590 0697 4  
591 P 0698 4 PARSE CALL(LOCAL TRUE FALSE, D_FLAG,  
592 0699 4 SJCS_FILE_FLAG, SJCS_NO_FILE_FLAG);  
593 0700 4  
594 0701 4  
595 0702 4 ! Get the /HEADER qualifier.  
596 0703 4  
597 P 0704 4 PARSE CALL(IF TRUE, D_HEADER,  
598 0705 4 SJCS_PAGE_HEADER);  
599 0706 4  
600 0707 4  
601 0708 4 ! Get the /PAGES qualifier.  
602 0709 4  
603 P 0710 4 PARSE CALL(LOWER UPPER, D_PAGES,  
604 P 0711 4 SJCS_FIRST_PAGE, SJCS_NO_FIRST_PAGE,  
605 0712 4 SJCS_LAST_PAGE, SJCS_NO_LAST_PAGE);  
606 0713 4  
607 0714 4  
608 0715 4 ! Get the /PASSALL qualifier.  
609 0716 4  
610 P 0717 4 PARSE CALL(IF TRUE, D_PASSALL,  
611 0718 4 SJCS_PASSALL);  
612 0719 4  
613 0720 4  
614 0721 4 ! Get the /SETUP qualifier.  
615 0722 4  
616 P 0723 4 PARSE CALL(COMMA LIST, D_SETUP,  
617 0724 4 SJCS_FILE_SETUP_MODULES, SJCS_NO_FILE_SETUP_MODULES);  
618 0725 4  
619 0726 4
```



```

: 620      0727 4      ! Get the /SPACE qualifier.
: 621      0728 4
: 622      P 0729 4      PARSE CALL(IF TRUE, D SPACE,
: 623      0730 4          SJCS_DOUBLE_SPACE);
: 624      0731 4
: 625      0732 4
: 626      0733 4      ! Get the local-level /TRAILER qualifier.
: 627      0734 4
: 628      P 0735 4      PARSE CALL(LOCAL TRUE FALSE, D TRAILER,
: 629      0736 4          SJCS_FILE_TRAILER, SJCS_NO_FILE_TRAILER);
: 630      0737 4      END;
: 631      0738 4
: 632      0739 4
: 633      0740 4      ! Loop, calling LIB$FIND_FILE to get files matching the input spec.
: 634      0741 4
: 635      0742 3      UNTIL
: 636      0743 4          BEGIN
: 637      0744 4              STATUS_2 = LIB$FIND_FILE(
: 638      0745 4                  INPUT_DESC,
: 639      0746 4                  FIND_RESULT,
: 640      0747 4                  FIND_CONTEXT,
: 641      0748 4                  FIND_DEFAULT,
: 642      0749 4                  0, 0, %REF(2));          !Request multiple input file processing
: 643      0750 4              .STATUS_2 EQL RMSS_NMF
: 644      0751 4          END
: 645      0752 3      DO
: 646      0753 4          BEGIN
: 647      0754 4              IF NOT .STATUS_2          ! Report error
: 648      0755 4              THEN
: 649      0756 5                  BEGIN
: 650      0757 5                      SIGNAL(
: 651      0758 6                          (IF .FLAGS[V PRINT]
: 652      0759 6                              THEN SUBMIT$FACILITY*16 + SHRS_OPENIN + STSSK_ERROR
: 653      0760 5                              ELSE SUBMIT$FACILITY*16 + SHRS_OPENIN + STSSK_SEVERE),
: 654      0761 5                          1, FIND_RESULT,
: 655      0762 5                          .FIND_CONTEXT[FAB$L_STS], .FIND_CONTEXT[FAB$L_STV]);
: 656      0763 5                  END
: 657      0764 4              ELSE
: 658      0765 5                  BEGIN
: 659      0766 5                      LOCAL
: 660      0767 5                          PROMPT_DESC;
: 661      0768 5
: 662      0769 5                          PROMPT_DESC = FIND_RESULT;          ! Pointer to a desc of
: 663      0770 5                              ! the filename
: 664      0771 5
: 665      0772 5                          STATUS_3 = LIB$QUAL FILE_MATCH (
: 666      0773 5                              QUAL_CONTEXT,          ! Qualifier context
: 667      0774 5                              FIND_CONTEXT,          ! Fab pointer
: 668      0775 5                              0,          ! No file name
: 669      0776 6                              ( IF .FLAGS[V PRINT]
: 670      0777 6                                  THEN $DESCRIPTOR('!AS, Print? [N]:')
: 671      0778 5                                  ELSE $DESCRIPTOR('!AS, Submit? [N]:')),
: 672      0779 5                              PROMPT_DESC,          ! Prompt
: 673      0780 5                              0);          ! Descriptor filename
: 674      0781 5                              ! No prompt routine
: 675      0782 5
: 676      0783 6                          IF NOT (.STATUS_3)
```

```
677      0784 5      THEN
678      0785 6      BEGIN
679      0786 6      IF (.STATUS_3 NEQ LIB$ FILFAIMAT) AND
680      0787 6      (.STATUS_3 NEQ LIB$ QUIPRO) AND
681      0788 7      (.STATUS_3 NEQ LIB$ NEGANS)
682      0789 6      THEN
683      0790 7      BEGIN
684      0791 7      SIGNAL (SUBMIT$ FACILITY^16 +
685      0792 7      SHR$ FICNOTACC + ST$SK_WARNING,
686      0793 7      1, FIND_RESULT);
687      0794 7      END
688      0795 7      ELSE IF (.STATUS_3 EQL LIB$ QUIPRO)
689      0796 6      THEN
690      0797 6      EXITLOOP;          ! Quit Now
691      0798 6      END
692      0799 5      ELSE
693      0800 6      BEGIN
694      0801 6      FIND_CONTEXT[FAB$V_SPL] = FIND_CONTEXT[FAB$V_SCF] = FALSE;
695      0802 6
696      0803 6      IF NOT $OPEN(FAB=.FIND_CONTEXT)
697      0804 7      THEN
698      0805 6      SIGNAL(
699      0806 6      (IF .FLAGS[V_PRINT]
700      0807 7      THEN SUBMIT$ FACILITY^16 + SHR$ OPENIN + ST$SK_ERROR
701      0808 7      ELSE SUBMIT$ FACILITY^16 + SHR$ OPENIN + ST$SK_SEVERE),
702      0809 6      1, FIND_RESULT,
703      0810 6      .FIND_CONTEXT[FAB$L_STS], .FIND_CONTEXT[FAB$L_STV])
704      0811 6      ELSE
705      0812 6      BEGIN
706      0813 7      IF .FLAGS[V_REMOTE]
707      0814 7      THEN
708      0815 7      BEGIN
709      0816 8      IF NOT .BBLOCK[FIND_CONTEXT[FAB$L_DEV], DEV$V_NET]
710      0817 8      THEN
711      0818 8      SIGNAL(
712      0819 8      SUBMIT$ REMOINSF,
713      0820 8      1, FIND_RESULT)
714      0821 8      ELSE
715      0822 8      IF .FLAGS[V_PRINT]
716      0823 8      THEN FIND_CONTEXT[FAB$V_SPL] = TRUE
717      0824 8      ELSE FIND_CONTEXT[FAB$V_SCF] = TRUE;
718      0825 8      END;
719      0826 7
720      0827 7      IF NOT $CLOSE(FAB=.FIND_CONTEXT)
721      0828 7      THEN
722      0829 8      SIGNAL(
723      0830 7      SUBMIT$ FACILITY^16 + SHR$ CLOSEIN + ST$SK_ERROR,
724      0831 7      1, FIND_RESULT,
725      0832 7      .FIND_CONTEXT[FAB$L_STS], .FIND_CONTEXT[FAB$L_STV]);
726      0833 7
727      0834 7      IF NOT .FLAGS[V_REMOTE]
728      0835 7      THEN
729      0836 7      BEGIN
730      0837 7      IF .BBLOCK[FIND_CONTEXT[FAB$L_DEV], DEV$V_NET]
```



734	0841	8
735	0842	8
736	0843	8
737	0844	8
738	0845	8
739	0846	9
740	0847	9
741	0848	9
742	0849	9
743	0850	9
744	0851	9
745	0852	9
746	0853	9
747	0854	9
748	0855	10
749	0856	10
750	0857	10
751	0858	10
752	0859	10
753	0860	10
754	0861	10
755	0862	10
756	0863	10
757	0864	10
758	0865	10
759	0866	10
760	0867	10
761	0868	10
762	0869	10
763	0870	10
764	0871	10
765	0872	10
766	0873	10
767	0874	10
768	0875	10
769	0876	10
770	0877	10
771	0878	10
772	0879	10
773	0880	10
774	0881	10
775	0882	10
776	0883	10
777	0884	10
778	0885	10
779	0886	10
780	0887	9
781	0888	8
782	0889	7
783	0890	6
784	0891	5
785	0892	4
786	0893	3
787	0894	2
788	0895	2
789	0896	2
790	0897	2

```

791 0898 BEGIN
792 0899 LOCAL
793 0900     IOSB:          VECTOR[4,WORD], ! I/O status block
794 0901     STATUS:      ! Status return
795 0902
796 0903
797 0904 ! Add the JOB_STATUS_OUTPUT item and finish the list.
798 0905
799 0906 Q_ICURSOR = ITEM_BUFFER;
800 0907 IF .FLAGS[V_IDENTIFY]
801 0908 THEN
802 0909     BEGIN
803 0910         JSM_DESC[0] = 0;
804 0911         JSM_DESC[1] = JSM_BUFFER;
805 0912         Q_ICURSOR[0,0,16,0] = %ALLOCATION(JSM_BUFFER);
806 0913         Q_ICURSOR[2,0,16,0] = SJC$ JOB_STATUS_OUTPUT;
807 0914         Q_ICURSOR[4,0,32,0] = JSM_BUFFER;
808 0915         Q_ICURSOR[8,0,32,0] = JSM_DESC;
809 0916         Q_ICURSOR = .Q_ICURSOR + T2;
810 0917     END;
811 0918 Q_ICURSOR[0,0,32,0] = 0;
812 0919
813 0920
814 0921 ! Send the close job request.
815 0922
816 0923 STATUS = $SNDJBCW(
817 0924     FUNC=SJC$ CLOSE JOB,
818 0925     ITMLST=ITEM_BUFFER,
819 0926     IOSB=IOSB);
820 0927
821 0928
822 0929 ! Cancel the exit handler established to delete the job on abnormal exit.
823 0930
824 0931 $CANEXH(DESBLOCK=EXIT_BLOCK);
825 0932
826 0933
827 0934 ! If close job failed and the count of jobs sent indicates that there
828 0935 ! WERE jobs then issue error.
829 0936
830 0937 IF .STATUS THEN STATUS = .IOSB;
831 0938 IF NOT .STATUS AND (.CNT NEQ 0)
832 0939 THEN
833 0940     SIGNAL STOP(
834 0941         SUBMIT$_CREJOB, 0, .STATUS);
835 0942
836 0943
837 0944 ! Execute /IDENTIFY only if there WAS a job. Assume an error
838 0945 ! was issued if no tasks in the job.
839 0946
840 0947 IF .FLAGS[V_IDENTIFY] AND (.CNT NEQ 0)
841 0948 THEN
842 0949     SIGNAL(
843 0950         SUBMIT$ FACILITY^16 + SHR$_TEXT + STS$K_SUCCESS,
844 0951         1 OR %B'0001'^16,
845 0952         JSM_DESC);
846 0953
847 0954 END;
```



SUBMIT  
V04-000

PRINT and SUBMIT Commands

M 11  
16-Sep-1984 00:28:51  
14-Sep-1984 12:09:55

VAX-11 Bliss-32 V4.0-742  
[CLIUTL.SRC]SUBMIT.B32;1

Page 17  
(3)

```
: 848
: 849
: 850
: 851
: 852

0955 2
0956 2 ! Return with the most severe error encountered.
0957 2 !
0958 2 .EXIT_STATUS OR ST$M_INHIB_MSG
0959 1 END;
```

.TITLE SUBMIT PRINT and SUBMIT Commands  
.IDENT \V04-000\

.PSECT DATA,NOEXE,2

00000 QUAL\_CONTEXT::  
          .BLK 4  
00004 EXIT\_STATUS:  
          .BLK 4  
00008 FLAGS: .BLK 4

.PSECT CODE,NOWRT,2

```
42 52 45 56 24 00000 P.AAB: .ASCII \SVERB\
                  00005 .BLK 3
                  00000005 00008 P.AAA: .LONG 5
                  00000000' 0000C .ADDRESS P.AAB
                  31 50 00010 P.AAD: .ASCII \P1\
                          00012 .BLK 2
                  00000002 00014 P.AAC: .LONG 2
                  00000000' 00018 .ADDRESS P.AAD
52 45 54 46 41 0001C P.AAF: .ASCII \AFTER\
                  00021 .BLK 3
                  00000005 00024 P.AAE: .LONG 5
                  00000000' 00028 .ADDRESS P.AAF
54 53 52 55 42 0002C P.AAH: .ASCII \BURST\
                  00031 .BLK 3
                  00000005 00034 P.AAG: .LONG 5
                  00000000' 00038 .ADDRESS P.AAH
53 43 49 54 53 49 52 45 54 43 41 52 41 48 43 0003C P.AAJ: .ASCII \CHARACTERISTICS\
                  0004B .BLK 1
                  0000000F 0004C P.AAI: .LONG 15
                  00000000' 00050 .ADDRESS P.AAJ
                  49 4C 43 00054 P.AAL: .ASCII \CLI\
                          00057 .BLK 1
                  00000003 00058 P.AAK: .LONG 3
                  00000000' 0005C .ADDRESS P.AAL
53 45 49 50 4F 43 00060 P.AAN: .ASCII \COPIES\
                  00066 .BLK 2
                  00000006 00068 P.AAM: .LONG 6
                  00000000' 0006C .ADDRESS P.AAN
45 4D 49 54 55 50 43 00070 P.AAP: .ASCII \CPUTIME\
                  00077 .BLK 1
                  00000007 00078 P.AAQ: .LONG 7
                  00000000' 0007C .ADDRESS P.AAP
45 54 45 4C 45 44 00080 P.AAR: .ASCII \DELETE\
                  00086 .BLK 2
                  00000006 00088 P.AAQ: .LONG 6
                  00000000' 0008C .ADDRESS P.AAR
45 43 49 56 45 44 00090 P.AAT: .ASCII \DEVICE\
```

LINE	ADDRESS	DATA	OPERATION	OPERAND	COMMENT
	00000006	00096			.BLKB 2
	00000000	00098	P.AAS:		.LONG 6
	00000000	0009C			.ADDRESS P.AAT
44	45 45 46	000A0	P.AAV:		.ASCII \FEED\
	00000004	000A4	P.AAU:		.LONG 4
	00000000	000A8			.ADDRESS P.AAV
47	41 4C 46	000AC	P.AAX:		.ASCII \FLAG\
	00000004	000B0	P.AAW:		.LONG 4
	00000000	000B4			.ADDRESS P.AAX
4D	52 4F 46	000B8	P.AAZ:		.ASCII \FORM\
	00000004	000BC	P.AAY:		.LONG 4
	00000000	000C0			.ADDRESS P.AAZ
52	45 44 41 45 48	000C4	P.ABB:		.ASCII \HEADER\
		000CA			.BLKB 2
	00000006	000CC	P.ABA:		.LONG 6
	00000000	000D0			.ADDRESS P.ABB
44	4C 4F 48	000D4	P.ABD:		.ASCII \HOLD\
	00000004	000D8	P.ABC:		.LONG 4
	00000000	000DC			.ADDRESS P.ABD
59	46 49 54 4E 45 44 49	000E0	P.ABF:		.ASCII \IDENTIFY\
	00000008	000E8	P.ABE:		.LONG 8
	00000000	000EC			.ADDRESS P.ABF
54	4E 55 4F 43 5F 42 4F 4A	000F0	P.ABH:		.ASCII \JOB_COUNT\
		000F9			.BLKB 3
	00000009	000FC	P.ABG:		.LONG 9
	00000000	00100			.ADDRESS P.ABH
50	45 45 4B	00104	P.ABJ:		.ASCII \KEEP\
	00000004	00108	P.ABI:		.LONG 4
	00000000	0010C			.ADDRESS P.ABJ
45	4C 49 46 5F 47 4F 4C	00110	P.ABL:		.ASCII \LOG_FILE\
	00000008	00118	P.ABK:		.LONG 8
	00000000	0011C			.ADDRESS P.ABL
45	53 41 43 52 45 57 4F 4C	00120	P.ABN:		.ASCII \LOWERCASE\
		00129			.BLKB 3
	00000009	0012C	P.ABM:		.LONG 9
	00000000	00130			.ADDRESS P.ABN
45	4D 41 4E	00134	P.ABP:		.ASCII \NAME\
	00000004	00138	P.ABO:		.LONG 4
	00000000	0013C			.ADDRESS P.ABP
45	54 4F 4E	00140	P.ABR:		.ASCII \NOTE\
	00000004	00144	P.ABQ:		.LONG 4
	00000000	00148			.ADDRESS P.ABR
59	46 49 54 4F 4E	0014C	P.ABT:		.ASCII \NOTIFY\
		00152			.BLKB 2
	00000006	00154	P.ABS:		.LONG 6
	00000000	00158			.ADDRESS P.ABT
52	4F 54 41 52 45 50 4F	0015C	P.ABV:		.ASCII \OPERATOR\
	00000008	00164	P.ABU:		.LONG 8
	00000000	00168			.ADDRESS P.ABV
53	45 47 41 50	0016C	P.ABX:		.ASCII \PAGES\
		00171			.BLKB 3
	00000005	00174	P.ABW:		.LONG 5
	00000000	00178			.ADDRESS P.ABX
53	52 45 54 45 4D 41 52 41 50	0017C	P.ABZ:		.ASCII \PARAMETERS\
		00186			.BLKB 2
	0000000A	00188	P.ABY:		.LONG 10
	00000000	0018C			.ADDRESS P.ABZ



4C	4C	41	53	53	41	50	00190	P.ACB:	.ASCII	\PASSALL\	:	
							00197		.BLKB	1	:	
						00000007	00198	P.ACA:	.LONG	7	:	
						00000000	0019C		.ADDRESS	P.ACB	:	
52	45	54	4E	49	52	50	001A0	P.ACD:	.ASCII	\PRINTER\	:	
							001A7		.BLKB	1	:	
						00000007	001A8	P.ACC:	.LONG	7	:	
						00000000	001AC		.ADDRESS	P.ACD	:	
59	54	49	52	4F	49	52	50	001B0	P.ACF:	.ASCII	\PRIORITY\	
						00000008	001B8	P.ACE:	.LONG	8	:	
						00000000	001BC		.ADDRESS	P.ACF	:	
		45	55	45	55	51	001C0	P.ACH:	.ASCII	\QUEUE\	:	
							001C5		.BLKB	3	:	
						00000005	001C8	P.ACG:	.LONG	5	:	
						00000000	001CC		.ADDRESS	P.ACH	:	
	45	54	4F	4D	45	52	001D0	P.ACJ:	.ASCII	\REMOTE\	:	
							001D6		.BLKB	2	:	
						00000006	001D8	P.ACI:	.LONG	6	:	
						00000000	001DC		.ADDRESS	P.ACJ	:	
54	52	41	54	53	45	52	001E0	P.ACL:	.ASCII	\RESTART\	:	
							001E7		.BLKB	1	:	
						00000007	001E8	P.ACK:	.LONG	7	:	
						00000000	001EC		.ADDRESS	P.ACL	:	
		50	55	54	45	53	001F0	P.ACN:	.ASCII	\SETUP\	:	
							001F5		.BLKB	3	:	
						00000005	001F8	P.ACM:	.LONG	5	:	
						00000000	001FC		.ADDRESS	P.ACN	:	
		45	43	41	50	53	00200	P.ACP:	.ASCII	\SPACE\	:	
							00205		.BLKB	3	:	
						00000005	00208	P.ACO:	.LONG	5	:	
						00000000	0020C		.ADDRESS	P.ACP	:	
52	45	4C	49	41	52	54	00210	P.ACR:	.ASCII	\TRAILER\	:	
							00217		.BLKB	1	:	
						00000007	00218	P.ACQ:	.LONG	7	:	
						00000000	0021C		.ADDRESS	P.ACR	:	
			52	45	53	55	00220	P.ACT:	.ASCII	\USER\	:	
						00000004	00224	P.ACS:	.LONG	4	:	
						00000000	00228		.ADDRESS	P.ACT	:	
54	4C	55	41	46	45	44	53	57	0022C	P.ACV:	.ASCII	\WSDEFAULT\
							00235		.BLKB	3	:	
						00000009	00238	P.ACU:	.LONG	9	:	
						00000000	0023C		.ADDRESS	P.ACV	:	
54	4E	45	54	58	45	53	57	00240	P.ACX:	.ASCII	\WSEXTENT\	
						00000008	00248	P.ACW:	.LONG	8	:	
						00000000	0024C		.ADDRESS	P.ACX	:	
		41	54	4F	55	51	53	57	00250	P.ACZ:	.ASCII	\WSQUOTA\
							00257		.BLKB	1	:	
						00000007	00258	P.ACY:	.LONG	7	:	
						00000000	0025C		.ADDRESS	P.ACZ	:	
4C	4C	41	2E	54	53	52	55	42	00260	P.ADB:	.ASCII	\BURST.ALL\
							00269		.BLKB	3	:	
						00000009	0026C	P.ADA:	.LONG	9	:	
						00000000	00270		.ADDRESS	P.ADB	:	
45	4E	4F	2E	54	53	52	55	42	00274	P.ADD:	.ASCII	\BURST.ONE\
							0027D		.BLKB	3	:	
						00000009	00280	P.ADC:	.LONG	9	:	
						00000000	00284		.ADDRESS	P.ADD	:	

```
      4C 4C 41 2E 47 41 4C 46 00288 P.ADF: .ASCII \FLAG.ALL\
      00000008 00290 P.ADE: .LONG 8
      00000000 00294 .ADDRESS P.ADF
      45 4E 4F 2E 47 41 4C 46 00298 P.ADH: .ASCII \FLAG.ONE\
      00000008 002A0 P.ADG: .LONG 8
      00000000 002A4 .ADDRESS P.ADH
      4C 4C 41 2E 52 45 4C 49 41 52 54 002A8 P.ADJ: .ASCII \TRAILER.ALL\
      0000000B 002B3 .BLKB 1
      00000000 002B4 P.ADI: .LONG 11
      00000000 002B8 .ADDRESS P.ADJ
      45 4E 4F 2E 52 45 4C 49 41 52 54 002BC P.ADL: .ASCII \TRAILER.ONE\
      0000000B 002C7 .BLKB 1
      00000000 002C8 P.ADK: .LONG 11
      00000000 002CC .ADDRESS P.ADL
      4D 4F 43 2E 002D0 P.ADM: .ASCII \.COM\
      53 49 4C 2E 002D4 P.ADN: .ASCII \.LIS\
      54 4E 49 52 50 24 53 59 53 002D8 P.ADP: .ASCII \SYSS$PRINT\
      00000009 002E1 .BLKB 3
      00000000 002E4 P.ADO: .LONG 9
      00000000 002E8 .ADDRESS P.ADP
      54 4E 49 52 50 24 53 59 53 002EC P.ADR: .ASCII \SYSS$PRINT\
      00000009 002F5 .BLKB 3
      00000000 002F8 P.ADQ: .LONG 9
      00000000 002FC .ADDRESS P.ADR
      47 4F 4C 2E 3A 4E 49 47 4F 4C 24 53 59 53 00300 P.ADT: .ASCII \SYSS$LOGIN:.LOG\
      0000000E 0030E .BLKB 2
      00000000 00310 P.ADS: .LONG 14
      00000000 00314 .ADDRESS P.ADT
      48 43 54 41 42 24 53 59 53 00318 P.ADV: .ASCII \SYSS$BATCH\
      00000009 00321 .BLKB 3
      00000000 00324 P.ADU: .LONG 9
      00000000 00328 .ADDRESS P.ADV
      5D 4E 5B 20 3F 74 6E 69 72 50 20 2C 53 41 21 0032C P.ADX: .ASCII \!AS, Print? [N]:\
      3A 0033B
      00000010 0033C P.ADW: .LONG 16
      00000000 00340 .ADDRESS P.ADX
      4E 5B 20 3F 74 69 6D 62 75 53 20 2C 53 41 21 00344 P.ADZ: .ASCII \!AS, Submit? [N]:\
      3A 5D 00353
      00000011 00355 .BLKB 3
      00000000 00358 P.ADY: .LONG 17
      00000000 0035C .ADDRESS P.ADZ
```

```
D-$VERB= P.AAA
D-P1= P.AAC
D-AFTER= P.AAE
D-BURST= P.AAG
D-CHARACTERISTICS= P.AAI
D-CLI= P.AAK
D-COPIES= P.AAM
D-CPUTIME= P.AAO
D-DELETE= P.AAQ
D-DEVICE= P.AAS
D-FEED= P.AAU
D-FLAG= P.AAW
D-FORM= P.AAY
D-HEADER= P.ABA
D-HOLD= P.ABC
```



D_IDENTIFY=	P.ABE
D_JOB_COUNT=	P.ABG
D_KEEP=	P.ABI
D_LOG_FILE=	P.ABK
D_LOWERCASE=	P.ABM
D_NAME=	P.ABO
D_NOTE=	P.ABQ
D_NOTIFY=	P.ABS
D_OPERATOR=	P.ABU
D_PAGES=	P.ABW
D_PARAMETERS=	P.ABY
D_PASSALL=	P.ACA
D_PRINTER=	P.ACC
D_PRIORITY=	P.ACE
D_QUEUE=	P.ACG
D_REMOTE=	P.ACI
D_RESTART=	P.ACK
D_SETUP=	P.ACM
D_SPACE=	P.ACO
D_TRAILER=	P.ACQ
D_USER=	P.ACS
D_WSDEFAULT=	P.ACU
D_WSEXTENT=	P.ACW
D_WSQUOTA=	P.ACY
D_BURST_ALL=	P.ADA
D_BURST_ONE=	P.ADC
D_FLAG_ALL=	P.ADE
D_FLAG_ONE=	P.ADG
D_TRAILER_ALL=	P.ADI
D_TRAILER_ONE=	P.ADK
.EXTRN	CLISGET VALUE, CLISPRESNT
.EXTRN	LIB\$FIND_FILE, LIB\$SCOPY_DXDX
.EXTRN	LIB\$QUAL_FILE_MATCH
.EXTRN	LIB\$QUAL_FILE_PARSE
.EXTRN	LIB\$QUIPRO, LIB\$NEGANS
.EXTRN	LIB\$FILFAMAT, PRINTS_FACILITY
.EXTRN	SUBMITS_FACILITY
.EXTRN	PRINTS_ADDFIL, SUBMITS_ADDFIL
.EXTRN	PRINTS_CREJOB, SUBMITS_CREJOB
.EXTRN	PRINTS_INVFILSPE
.EXTRN	SUBMITS_INVFILSPE
.EXTRN	PRINTS_REMOINSF
.EXTRN	SUBMITS_REMOINSF
.EXTRN	SUBMITS_INVLOGFIL
.EXTRN	PARSE_AFTER, PARSE_CHARACTERISTICS
.EXTRN	PARSE_IF_TRUE, PARSE_PARAMETERS
.EXTRN	PARSE_PRIORITY, PARSE_IF_TRUE_FALSE
.EXTRN	PARSE_USER, PARSE_IF_FALSE
.EXTRN	PARSE_QUEUE, PARSE_FORM
.EXTRN	PARSE_COPIES, PARSE_NAME
.EXTRN	PARSE_STRING, PARSE_FILENAME
.EXTRN	PARSE_CPUTIME, PARSE_LOG_FILE
.EXTRN	PARSE_NAME_AND_LOG_FILE
.EXTRN	PARSE_PRINTER, PARSE_WORKING_SET
.EXTRN	SYSSNDJBCW, SYSSDCLEXH
.EXTRN	PARSE_LOCAL_TRUE_FALSE
.EXTRN	PARSE_LOWER_UPPER

0271

Address	Hex	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	Op418
---------	-----	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------



		0150	C6	9F	000F2	PUSHAB	D RESTART	:	
0000G	CF		03	FB	000F6	CALLS	#3, PARSE_IF_TRUE_FALSE	:	
		018C	C6	9F	000FB	PUSHAB	D USER	:	0473
0000G	CF		01	FB	000FF	CALLS	#T, PARSE_USER	:	
	03		67	E8	00104	BLBS	FLAGS, 2\$	:	0476
			00F6	31	00107	BRW	5\$	:	
			20	DD	0010A	PUSHL	#32	:	0482
		01D4	C6	9F	0010C	PUSHAB	D BURST ALL	:	
0000G	CF		02	FB	00110	CALLS	#2, PARSE_IF_TRUE	:	
			21	DD	00115	PUSHL	#33	:	0483
		01E8	C6	9F	00117	PUSHAB	D BURST ONE	:	
0000G	CF		02	FB	0011B	CALLS	#2, PARSE_IF_TRUE	:	
			22	DD	00120	PUSHL	#34	:	0484
		9C	A6	9F	00122	PUSHAB	D BURST	:	
0000G	CF		02	FB	00125	CALLS	#2, PARSE_IF_FALSE	:	
			56	DD	0012A	PUSHL	R6	:	0489
00000000G	00		01	FB	0012C	CALLS	#1, CLISPRESNT	:	
	1D		50	E9	00133	BLBC	R0, 3\$	:	
		0130	C6	9F	00136	PUSHAB	D QUEUE	:	
00000000G	00		01	FB	0013A	CALLS	#T, CLISPRESNT	:	
	0F		50	E8	00141	BLBS	R0, 3\$	:	
		024C	C6	9F	00144	PUSHAB	P.ADO	:	0491
	7E	86	8F	9A	00148	MOVZBL	#134, -(SP)	:	
			56	DD	0014C	PUSHL	R6	:	
0000G	CF		03	FB	0014E	CALLS	#3, PARSE_QUEUE	:	
			24	DD	00153	PUSHL	#36	:	0496
		01F8	C6	9F	00155	PUSHAB	D FLAG ALL	:	
0000G	CF		02	FB	00159	CALLS	#2, PARSE_IF_TRUE	:	
			25	DD	0015E	PUSHL	#37	:	0497
		0208	C6	9F	00160	PUSHAB	D FLAG ONE	:	
0000G	CF		02	FB	00164	CALLS	#2, PARSE_IF_TRUE	:	
			26	DD	00169	PUSHL	#38	:	0498
		18	A6	9F	0016B	PUSHAB	D FLAG	:	
0000G	CF		02	FB	0016E	CALLS	#2, PARSE_IF_FALSE	:	
		24	A6	9F	00173	PUSHAB	D FORM	:	0503
0000G	CF		01	FB	00176	CALLS	#T, PARSE_FORM	:	
	7E	4B	8F	9A	0017B	MOVZBL	#75, -(SP)	:	0509
		64	A6	9F	0017F	PUSHAB	D JOB COUNT	:	
0000G	CF		02	FB	00182	CALLS	#2, PARSE_COPIES	:	
	7E	66	8F	9A	00187	MOVZBL	#102, -(SP)	:	0515
		0094	C6	9F	0018B	PUSHAB	D LOWERCASE	:	
0000G	CF		02	FB	0018F	CALLS	#2, PARSE_IF_TRUE	:	
		00A0	C6	9F	00194	PUSHAB	D NAME	:	0520
0000G	CF		01	FB	00198	CALLS	#T, PARSE_NAME	:	
	7E	6B	8F	9A	0019D	MOVZBL	#107, -(SP)	:	0526
	7E	6A	8F	9A	001A1	MOVZBL	#106, -(SP)	:	
		00AC	C6	9F	001A5	PUSHAB	D NOTE	:	
0000G	CF		03	FB	001A9	CALLS	#3, PARSE_STRING	:	
	7E	6F	8F	9A	001AE	MOVZBL	#111, -(SP)	:	0532
	7E	6E	8F	9A	001B2	MOVZBL	#110, -(SP)	:	
		00CC	C6	9F	001B6	PUSHAB	D OPERATOR	:	
0000G	CF		03	FB	001BA	CALLS	#3, PARSE_STRING	:	
			56	DD	001BF	PUSHL	R6	:	0537
00000000G	00		01	FB	001C1	CALLS	#1, CLISPRESNT	:	
	11		50	E8	001C8	BLBS	R0, 4\$	:	
		0260	C6	9F	001CB	PUSHAB	P.ADO	:	0539
	7E	86	8F	9A	001CF	MOVZBL	#134, -(SP)	:	

0000G	CF	0130	C6	9F	001D3	PUSHAB	D QUEUE	...	
			03	FB	001D7	CALLS	#3, PARSE_QUEUE	...	0544
			2B	DD	001DC	PUSHL	#43	...	
0000G	CF	021C	C6	9F	001DE	PUSHAB	D TRAILER_ALL	...	0545
			02	FB	001E2	CALLS	#2, PARSE_IF_TRUE	...	
			2C	DD	001E7	PUSHL	#44	...	
0000G	CF	0230	C6	9F	001E9	PUSHAB	D TRAILER_ONE	...	0546
			02	FB	001ED	CALLS	#2, PARSE_IF_TRUE	...	
			2D	DD	001F2	PUSHL	#45	...	
0000G	CF	0180	C6	9F	001F4	PUSHAB	D TRAILER	...	0476
			02	FB	001F8	CALLS	#2, PARSE_IF_FALSE	...	0554
			00A9	31	001FD	BRW	7\$	...	
			12	DD	00200	PUSHL	#18	...	
			11	DD	00202	PUSHL	#17	...	
0000G	CF	C0	A6	9F	00204	PUSHAB	D CLI	...	0560
			03	FB	00207	CALLS	#3, PARSE_FILENAME	...	
			16	DD	0020C	PUSHL	#22	...	
			15	DD	0020E	PUSHL	#21	...	
0000G	CF	E0	A6	9F	00210	PUSHAB	D CPUTIME	...	0566
	7E	60	03	FB	00213	CALLS	#3, PARSE_CPUTIME	...	
		70	8F	9A	00218	MOVZBL	#96, -(SP)	...	
0000G	CF		A6	9F	0021C	PUSHAB	D KEEP	...	0571
	58		02	FB	0021F	CALLS	#2, PARSE_IF_TRUE	...	0572
		D8	AD	9E	00224	MOVAB	LOG FILE DESC, Q_VALUE_DESC	...	
0000G	CF	0080	C6	9F	00228	PUSHAB	D LOG FILE	...	0577
			01	FB	0022C	CALLS	#T, PARSE_LOG_FILE	...	0578
			50	FO	00231	INSV	RO, #3, #T, FLAGS	...	
			AD	9E	00236	MOVAB	VALUE DESC, Q_VALUE_DESC	...	0583
0000G	CF	00A0	C6	9F	0023A	PUSHAB	D NAME	...	0587
	67		01	FB	0023E	CALLS	#T, PARSE_NAME	...	
			03	E0	00243	BBS	#3, FLAGS, 6\$	...	
		00000000G	8F	DD	00247	PUSHL	#SUBMITS_INVLOGFIL	...	
		0278	C6	9F	0024D	PUSHAB	P.ADS	...	
		D8	AD	9F	00251	PUSHAB	LOG FILE DESC	...	
		E8	AD	9F	00254	PUSHAB	VALUE DESC	...	
0000G	CF		04	FB	00257	CALLS	#4, PARSE_NAME_AND_LOG_FILE	...	0592
0000G	CF	0110	C6	9F	0025C	PUSHAB	D PRINTER	...	0597
			01	FB	00260	CALLS	#T, PARSE_PRINTER	...	
		028C	C6	9F	00265	PUSHAB	P.ADU	...	
	7E	86	8F	9A	00269	MOVZBL	#134, -(SP)	...	
0000G	CF	0130	C6	9F	0026D	PUSHAB	D QUEUE	...	0603
	7E		03	FB	00271	CALLS	#3, PARSE_QUEUE	...	
	7E	98	8F	9A	00276	MOVZBL	#152, -(SP)	...	
		97	8F	9A	0027A	MOVZBL	#151, -(SP)	...	
0000G	CF	01A0	C6	9F	0027E	PUSHAB	D WSDFAULT	...	0609
			03	FB	00282	CALLS	#3, PARSE_WORKING_SET	...	
	7E	9A	8F	9A	00287	MOVZBL	#154, -(SP)	...	
	7E	99	8F	9A	0028B	MOVZBL	#153, -(SP)	...	
0000G	CF	0180	C6	9F	0028F	PUSHAB	D WSEXTENT	...	0615
			03	FB	00293	CALLS	#3, PARSE_WORKING_SET	...	
	7E	9C	8F	9A	00298	MOVZBL	#156, -(SP)	...	
	7E	9B	8F	9A	0029C	MOVZBL	#155, -(SP)	...	
0000G	CF	01C0	C6	9F	002A0	PUSHAB	D WSQUOTA	...	0621
	67		03	FB	002A4	CALLS	#3, PARSE_WORKING_SET	...	0631
			02	E0	002A9	BBS	#2, FLAGS, 10\$	...	0635
			6B	D4	002AD	CLRL	(Q ICURSOR)	...	
			7E	7C	002AF	CLRQ	-(SP)	...	



		14	AE	9F	002B1	PUSHAB	IOSB		
		0434	CE	9F	002B4	PUSHAB	ITEM_BUFFER		
	7E		0A	7D	002B8	MOVQ	#10, -(SP)		
			7E	D4	002BB	CLRL	-(SP)		
00000000G	00		07	FB	002BD	CALLS	#7, SYSSNDJBCW		
	07		50	E9	002C4	BLBC	STATUS, 8\$		0636
	50	0C	AE	D0	002C7	MOVL	IOSB, STATUS		
	11		50	E8	002CB	BLBS	STATUS, 9\$		0637
			50	DD	002CE	PUSHL	STATUS		0640
			7E	D4	002D0	CLRL	-(SP)		0639
		00000000G	8F	DD	002D2	PUSHL	#SUBMITS CREJOB		
00000000G	00		03	FB	002D8	CALLS	#3, LIB\$STOP		
		14	AE	D4	002DF	CLRL	EXIT_BLOCK		0645
	18	AE	0000V	CF	9E	MOVAB	EXIT_HANDLER, EXIT_BLOCK+4		0646
	1C	AE		01	D0	MOVL	#1, EXIT_BLOCK+8		0647
	20	AE		AE	9E	MOVAB	EXIT_BLOCK+16, EXIT_BLOCK+12		0648
		24		AE	D4	CLRL	EXIT_BLOCK+16		0649
		24		AE	9F	PUSHAB	EXIT_BLOCK		0650
		14		01	FB	CALLS	#1, SYSSDCLEXH		
00000000G	00		E0	AD	9F	PUSHAB	INPUT_DESC		0656
		FF7C	C6	9F	00301	PUSHAB	D_P1		
00000000G	00		02	FB	00305	CALLS	#2, CLISGET_VALUE		
	03		50	E8	0030C	BLBS	R0, 11\$		
		0236	31	0030F	BRW	35\$			
	5B	0428	CE	9E	00312	MOVAB	ITEM_BUFFER, Q_ICURSOR		0664
	5A	28	AE	9E	00317	MOVAB	DATA_BUFFER, Q_DCURSOR		0665
			18	DD	0031B	PUSHL	#24		0671
		F0	A6	9F	0031D	PUSHAB	D_DELETE		
0000G	CF		02	FB	00320	CALLS	#2, PARSE_IF_TRUE		
	03		67	E8	00325	BLBS	FLAGS, 12\$		0674
		0085	31	00328	BRW	13\$			
			22	DD	0032B	PUSHL	#34		0681
			20	DD	0032D	PUSHL	#32		
		9C	A6	9F	0032F	PUSHAB	D_BURST		
0000G	CF		03	FB	00332	CALLS	#3, PARSE_LOCAL_TRUE_FALSE		
			23	DD	00337	PUSHL	#35		0687
		D0	A6	9F	00339	PUSHAB	D_COPIES		
0000G	CF		02	FB	0033C	CALLS	#2, PARSE_COPIES		
	7E	76	8F	9A	00341	MOVZBL	#118, -(SP)		0693
	7E	75	8F	9A	00345	MOVZBL	#117, -(SP)		
		0C	A6	9F	00349	PUSHAB	D_FEED		
0000G	CF		03	FB	0034C	CALLS	#3, PARSE_IF_TRUE_FALSE		
			26	DD	00351	PUSHL	#38		0699
			24	DD	00353	PUSHL	#36		
		18	A6	9F	00355	PUSHAB	D_FLAG		
0000G	CF		03	FB	00358	CALLS	#3, PARSE_LOCAL_TRUE_FALSE		
	7E	71	8F	9A	0035D	MOVZBL	#113, -(SP)		0705
		34	A6	9F	00361	PUSHAB	D_HEADER		
0000G	CF		02	FB	00364	CALLS	#2, PARSE_IF_TRUE		
	7E	5C	8F	9A	00369	MOVZBL	#92, -(SP)		0712
	7E	5B	8F	9A	0036D	MOVZBL	#91, -(SP)		
			2F	DD	00371	PUSHL	#47		
			2E	DD	00373	PUSHL	#46		
		00DC	C6	9F	00375	PUSHAB	D_PAGES		
0000G	CF		05	FB	00379	CALLS	#5, PARSE_LOWER_UPPER		
	7E	80	8F	9A	0037E	MOVZBL	#128, -(SP)		0718
		0100	C6	9F	00382	PUSHAB	D_PASSALL		

0000G	CF		02	FB	00386	CALLS	#2, PARSE_IF_TRUE		
			29	DD	0038B	PUSHL	#41		0724
			28	DD	0038D	PUSHL	#40		
		0160	C6	9F	0038F	PUSHAB	D SETUP		
0000G	CF		03	FB	00393	CALLS	#3, PARSE_COMMA_LIST		0730
			1C	DD	00398	PUSHL	#28		
		0170	C6	9F	0039A	PUSHAB	D SPACE		
0000G	CF		02	FB	0039E	CALLS	#2, PARSE_IF_TRUE		0736
			2D	DD	003A3	PUSHL	#45		
			2B	DD	003A5	PUSHL	#43		
		0180	C6	9F	003A7	PUSHAB	D TRAILER		
0000G	CF		03	FB	003AB	CALLS	#3, PARSE_LOCAL_TRUE_FALSE		0749
	6E		02	DD	003B0	MOVL	#2, (SP)		
			5E	DD	003B3	PUSHL	SP		0744
			7E	7C	003B5	CLRG	-(SP)		
		F0	AD	9F	003B7	PUSHAB	FIND_DEFAULT		
		14	AE	9F	003BA	PUSHAB	FIND_CONTEXT		
		F8	AD	9F	003BD	PUSHAB	FIND_RESULT		
		E0	AD	9F	003C0	PUSHAB	INPUT_DESC		
00000000G	00		07	FB	003C3	CALLS	#7, LIB\$FIND_FILE		
	55		50	DD	003CA	MOVL	R0, STATUS_2		0750
000182CA	8F		55	D1	003CD	CMPL	STATUS_2, #99018		
			68	13	003D4	BEQL	17\$		0762
	52	04	AE	DD	003D6	MOVL	FIND_CONTEXT, R2		0758
	75		55	E9	003DA	BLBC	STATUS_2, 19\$		0769
08	AE	F8	AD	9E	003DD	MOVAB	FIND_RESULT, PROMPT_DESC		0772
			7E	D4	003E2	CLRL	-(SP)		
		0C	AE	9F	003E4	PUSHAB	PROMPT_DESC		0776
	07		67	E9	003E7	BLBC	FLAGS, 14\$		0777
	50	02A4	C6	9E	003EA	MOVAB	P.ADW, R0		
			05	11	003EF	BRB	15\$		0778
	50	02C0	C6	9E	003F1	MOVAB	P.ADY, R0		
			50	DD	003F6	PUSHL	R0		0772
			7E	D4	003F8	CLRL	-(SP)		0774
			52	DD	003FA	PUSHL	R2		0772
		F8	A7	9F	003FC	PUSHAB	QUAL_CONTEXT		
00000000G	00		06	FB	003FF	CALLS	#6, LIB\$QUAL_FILE_MATCH		
	53		50	DD	00406	MOVL	R0, STATUS_3		0783
	35		53	E8	00409	BLBS	STATUS_3, 18\$		0786
00000000G	8F		53	D1	0040C	CMPL	STATUS_3, #LIB\$FILFAIMAT		
			20	13	00413	BEQL	16\$		0787
00000000G	8F		53	D1	00415	CMPL	STATUS_3, #LIB\$QUIPRO		0788
			17	13	0041C	BEQL	16\$		
00000000G	8F		53	D1	0041E	CMPL	STATUS_3, #LIB\$NEGANS		
			0E	13	00425	BEQL	16\$		0791
		F8	AD	9F	00427	PUSHAB	FIND_RESULT		
			01	DD	0042A	PUSHL	#1		0792
		00000000*	8F	DD	0042C	PUSHL	#<<SUBMIT\$_FACILITY@16>+4920>		
			00BD	31	00432	BRW	30\$		0795
00000000G	8F		53	D1	00435	CMPL	STATUS_3, #LIB\$QUIPRO		
			35	12	0043C	BNEQ	22\$		0797
		FEBD	31	0043E	BRW	10\$			0802
05	A2	60	8F	8A	00441	BICB2	#96, 5(R2)		0804
			52	DD	00446	PUSHL	R2		
00000000G	00		01	FB	00448	CALLS	#1, SYSS\$OPEN		
	24		50	E8	0044F	BLBS	R0, 23\$		
	7E	08	A2	7D	00452	MOVQ	8(R2), -(SP)		0811



			F8	AD	9F	00456	PUSHAB	FIND_RESULT	0806
				01	DD	00459	PUSHL	#1	
	08			67	E9	0045B	BLBC	FLAGS, 20\$	0807
		00000000*		8F	DD	0045E	PUSHL	#<<<SUBMIT\$_FACILITY@16>+4248>+2>	0808
				06	11	00464	BRB	21\$	
		00000000*		8F	DD	00466	PUSHL	#<<<SUBMIT\$_FACILITY@16>+4248>+4>	0809
	00000000G	00		05	FB	0046C	CALLS	#5, LIB\$SIGNAL	0807
			FF	3A	31	00473	BRW	13\$	0806
27		67		02	E1	00476	BBC	#2, FLAGS, 26\$	0814
14		A2		05	E0	0047A	BBS	#5, 65(R2), 24\$	0817
			F8	AD	9F	0047F	PUSHAB	FIND_RESULT	0819
				01	DD	00482	PUSHL	#1	
		00000000G		8F	DD	00484	PUSHL	#SUBMIT\$_REMOINSF	
	00000000G	00		03	FB	0048A	CALLS	#3, LIB\$SIGNAL	
				0E	11	00491	BRB	26\$	
		06		67	E9	00493	BLBC	FLAGS, 25\$	0823
	05	A2		20	88	00496	BISB2	#32, 5(R2)	0824
				05	11	0049A	BRB	26\$	
	05	A2	40	8F	88	0049C	BISB2	#64, 5(R2)	0825
				52	DD	004A1	PUSHL	R2	0829
	00000000G	00		01	FB	004A3	CALLS	#1, SYS\$CLOSE	
		16		50	E8	004AA	BLBS	R0, 27\$	
		7E		AD	7D	004AD	MOVQ	8(R2), -(SP)	0834
			F8	AD	9F	004B1	PUSHAB	FIND_RESULT	0831
				01	DD	004B4	PUSHL	#1	
		00000000*		8F	DD	004B6	PUSHL	#<<<SUBMIT\$_FACILITY@16>+4176>+2>	0832
	00000000G	00		05	FB	004BC	CALLS	#5, LIB\$SIGNAL	
AC		67		02	E0	004C3	BBS	#2, FLAGS, 22\$	0837
OD		A2		05	E1	004C7	BBC	#5, 65(R2), 28\$	0840
			F8	AD	9F	004CC	PUSHAB	FIND_RESULT	0842
				01	DD	004CF	PUSHL	#1	
		00000000G		8F	DD	004D1	PUSHL	#SUBMIT\$_REMOINSF	
				19	11	004D7	BRB	30\$	
09		A2		06	E1	004D9	BBC	#6, 65(R2), 29\$	0847
	41			A2	E8	004DE	BLBS	67(R2), 29\$	0848
		05	43						
14		A2		05	E1	004E2	BBC	#5, 64(R2), 31\$	0849
	40								
			F8	AD	9F	004E7	PUSHAB	FIND_RESULT	0851
				01	DD	004EA	PUSHL	#1	
		00000000G		8F	DD	004EC	PUSHL	#SUBMIT\$_INVFILSPE	
				03	FB	004F2	CALLS	#3, LIB\$SIGNAL	
				4A	11	004F9	BRB	34\$	
		50	28	A2	D0	004FB	MOVL	40(R2), NAM	0864
		6B	0027001C	8F	D0	004FF	MOVL	#2555932, (Q_ICURSOR)	0866
	04	AB		A0	9E	00506	MOVAB	20(R0), 4(Q_ICURSOR)	0868
				AB	7C	0050B	CLRQ	8(Q_ICURSOR)	0869
				7E	7C	0050E	CLRQ	-(SP)	0878
		14		AE	9F	00510	PUSHAB	IOSB	
		0434		CE	9F	00513	PUSHAB	ITEM_BUFFER	
				02	7D	00517	MOVQ	#2, -(SP)	
				7E	D4	0051A	CLRL	-(SP)	
	00000000G	00		07	FB	0051C	CALLS	#7, SYS\$SNDJBCW	
		07		50	E9	00523	BLBC	STATUS, 32\$	0879
		50	0C	AE	D0	00526	MOVL	IOSB, STATUS	
		16		50	E8	0052A	BLBS	STATUS, 33\$	0880
				50	DD	0052D	PUSHL	STATUS	0884
			F8	AD	9F	0052F	PUSHAB	FIND_RESULT	0882
				01	DD	00532	PUSHL	#1	

00000000G	00	00000000G	8F	DD	00534	PUSHL	#SUBMIT\$ ADDFIL		
			04	FB	0053A	CALLS	#4, LIB\$SIGNAL		
			02	11	00541	BRB	34\$		
			54	D6	00543	INCL	CNT		0886
03	67		FE68	31	00545	BRW	13\$		0837
			02	E1	00548	BBC	#2, FLAGS, 36\$		0896
			0087	31	0054C	BRW	40\$		
	5B	0428	CE	9E	0054F	MOVAB	ITEM BUFFER, Q_ICURSOR		0906
1C	67		01	E1	00554	BBC	#1, FLAGS, 37\$		0907
		FF4C	CD	D4	00558	CLRL	JSM_DESC		0910
	FF50	FF54	CD	9E	0055C	MOVAB	JSM_BUFFER, JSM_DESC+4		0911
	8B	00580084	8F	D0	00563	MOVL	#5767300, (Q_ICURSOR)+		0912
	8B	FF54	CD	9E	0056A	MOVAB	JSM_BUFFER, (Q_ICURSOR)+		0914
	8B	FF4C	CD	9E	0056F	MOVAB	JSM_DESC, (Q_ICURSOR)+		0915
			6B	D4	00574	CLRL	(Q_ICURSOR)		0918
			7E	7C	00576	CLRQ	-(SP)		0926
		14	AE	9F	00578	PUSHAB	IOSB		
		0434	CE	9F	0057B	PUSHAB	ITEM_BUFFER		
	7E		09	7D	0057F	MOVQ	#9, -(SP)		
			7E	D4	00582	CLRL	-(SP)		
00000000G	00		07	FB	00584	CALLS	#7, SYS\$SNDJBCW		
	53		50	D0	0058B	MOVL	R0, STATUS		
		14	AE	9F	0058E	PUSHAB	EXIT_BLOCK		0931
00000000G	00		01	FB	00591	CALLS	#1, SYS\$SCANEXH		
	07		53	E9	00598	BLBC	STATUS, 38\$		0937
	53	0C	AE	D0	0059B	MOVL	IOSB, STATUS		
	15		53	E8	0059F	BLBS	STATUS, 39\$		0938
			54	D5	005A2	TSTL	CNT		
			11	13	005A4	BEQL	39\$		
			53	DD	005A6	PUSHL	STATUS		0941
			7E	D4	005A8	CLRL	-(SP)		0940
		00000000G	8F	DD	005AA	PUSHL	#SUBMIT\$ CREJOB		
00000000G	00		03	FB	005B0	CALLS	#3, LIB\$STOP		
1B	67		01	E1	005B7	BBC	#1, FLAGS, 40\$		0947
			54	D5	005BB	TSTL	CNT		
			17	13	005BD	BEQL	40\$		
		FF4C	CD	9F	005BF	PUSHAB	JSM_DESC		0949
		00010001	8F	DD	005C3	PUSHL	#65537		0951
		00000000*	8F	DD	005C9	PUSHL	#<<<SUBMIT\$ FACILITY@16>+4400>+1>		0950
00000000G	00		03	FB	005CF	CALLS	#3, LIB\$SIGNAL		
50	FC	A7 10000000	8F	C9	005D6	BISL3	#268435456, EXIT_STATUS, R0		0958
			04	005DF	RET				0959

; Routine Size: 1504 bytes, Routine Base: CODE + 0360



```
854 0960 1 ROUTINE GET_COMM_QUAL_CONTEXT: NOVALUE =
855 0961 1 ++
856 0962 1
857 0963 1 FUNCTIONAL DESCRIPTION:
858 0964 1 This routine is gets the common qualifier context.
859 0965 1
860 0966 1 INPUT PARAMETERS:
861 0967 1 NONE
862 0968 1
863 0969 1 IMPLICIT INPUTS:
864 0970 1 QUAL_CONTEXT - defined locally in main routine.
865 0971 1
866 0972 1 OUTPUT PARAMETERS:
867 0973 1 NONE
868 0974 1
869 0975 1 IMPLICIT OUTPUTS:
870 0976 1 NONE
871 0977 1
872 0978 1 ROUTINE VALUE:
873 0979 1 NONE
874 0980 1
875 0981 1 SIDE EFFECTS:
876 0982 1 NONE
877 0983 1
878 0984 1 --
879 0985 1
880 0986 2 BEGIN
881 0987 2
882 0988 2 LOCAL
883 0989 2 BITMAP,
884 0990 2 STATUS;
885 0991 2
886 0992 2 ! Set up the context bitmap
887 0993 2
888 0994 2 BITMAP= LIBSM_CQF_CONFIRM OR LIBSM_CQF_EXCLUDE OR
889 0995 2 LIBSM_CQF_BEFORE OR LIBSM_CQF_SINCE OR
890 0996 2 LIBSM_CQF_CREATED OR LIBSM_CQF_MODIFIED OR
891 0997 2 LIBSM_CQF_EXPIRED OR LIBSM_CQF_BACKUP OR
892 0998 2 LIBSM_CQF_BYOWNER;
893 0999 2
894 1000 2 ! Get the common qualifier context
895 1001 2
896 1002 3 IF NOT (STATUS = LIB$QUAL_FILE_PARSE (BITMAP, QUAL_CONTEXT))
897 1003 2 THEN
898 1004 2 SIGNAL_STOP (.STATUS);
899 1005 2
900 1006 1 END;
```

```
0000 00000 GET_COMM_QUAL_CONTEXT:
7E 01FF 8F 3C 00002 .WORD Save nothing
0000 04 CF 9F 00007 MOVZWL #511, BITMAP
04 AE 9F 0000B PUSHAB QUAL_CONTEXT
PUSHAB BITMAP
```

```
: 0960
: 0997
: 1002
:
```

## PRINT and SUBMIT Commands

M 12  
16-Sep-1984 00:28:51  
14-Sep-1984 12:09:55

VAX-11 BLISS-32 V4.0-742  
[CLIUTL.SRC]SUBMIT.B32;1

Page 30  
(4)

00000000G 00  
09

```
02 FB 0000E
50 E8 00015
50 DD 00018
01 FB 0001A
    04 00021 1$:
```

```
CALLS      #2, LIB$QUAL_FILE_PARSE
BLBS       STATUS, 1$
PUSHL      STATUS
CALLS      #1, LIB$STOP
RET
```

1004  
1006

00000000G 00

50 DD 00018  
01 FB 0001A  
04 00021 15:

PUSHL STATUS  
CALLS #1, LIB\$STOP  
RET

1004  
1006

; Routine Size: 34 bytes, Routine Base: CODE + 0940



```
: 902      1007 1 ROUTINE EXIT_HANDLER: NOVALUE=
: 903      1008 1
: 904      1009 1 ++
: 905      1010 1
: 906      1011 1 FUNCTIONAL DESCRIPTION:
: 907      1012 1 This routine is an exit handler that is established as long as the
: 908      1013 1 process owns an open job. Its function is to delete the open job
: 909      1014 1 if an image exit should occur.
: 910      1015 1
: 911      1016 1 INPUT PARAMETERS:
: 912      1017 1 Standard exit handler parameters (not used).
: 913      1018 1
: 914      1019 1 IMPLICIT INPUTS:
: 915      1020 1 NONE
: 916      1021 1
: 917      1022 1 OUTPUT PARAMETERS:
: 918      1023 1 NONE
: 919      1024 1
: 920      1025 1 IMPLICIT OUTPUTS:
: 921      1026 1 NONE
: 922      1027 1
: 923      1028 1 ROUTINE VALUE:
: 924      1029 1 NONE
: 925      1030 1
: 926      1031 1 SIDE EFFECTS:
: 927      1032 1 NONE
: 928      1033 1
: 929      1034 1 --
: 930      1035 1
: 931      1036 2 BEGIN
: 932      1037 2 $$SNDJBC(FUNC=SJC$_CLOSE_DELETE);
: 933      1038 1 END;
```

.EXTRN SYS\$\$SNDJBC

```
0000 00000 EXIT_HANDLER:
7E 7C 00002 .WORD Save nothing
7E 7C 00004 CLRQ -(SP)
08 7D 00006 CLRQ -(SP)
7E D4 00009 MOVQ #8, -(SP)
07 FB 0000B CLRL -(SP)
04 00012 CALLS #7, SYS$$SNDJBC
RET
```

```
: 1007
: 1037
:
:
:
: 1038
```

; Routine Size: 19 bytes, Routine Base: CODE + 0962

```

935 1039 1 ROUTINE HANDLER(SIG,MECH)=
936 1040 1
937 1041 1 ++
938 1042 1
939 1043 1 FUNCTIONAL DESCRIPTION:
940 1044 1 This routine is the condition handler for the main routine. It
941 1045 1 saves the most severe condition as the exit status.
942 1046 1
943 1047 1 INPUT PARAMETERS:
944 1048 1 Standard VMS condition handler parameters.
945 1049 1
946 1050 1 IMPLICIT INPUTS:
947 1051 1 NONE
948 1052 1
949 1053 1 OUTPUT PARAMETERS:
950 1054 1 NONE
951 1055 1
952 1056 1 IMPLICIT OUTPUTS:
953 1057 1 EXIT_STATUS - Contains the most severe status encountered.
954 1058 1
955 1059 1 ROUTINE VALUE:
956 1060 1 SSS_RESIGNAL
957 1061 1
958 1062 1 SIDE EFFECTS:
959 1063 1 NONE
960 1064 1
961 1065 1 --
962 1066 1
963 1067 2 BEGIN
964 1068 2 MAP
965 1069 2 SIG: REF BBLOCK, ! Signal parameters
966 1070 2 MECH: REF BBLOCK; ! Mechanism parameters
967 1071 2 BIND
968 1072 2 SIGNAME = SIG[CHF$$_SIG_NAME] : BBLOCK; ! Name of signal
969 1073 2
970 1074 2
971 1075 2 ! Change the facility code from SUBMIT to PRINT if required.
972 1076 2
973 1077 2 IF
974 1078 2 .FLAGS[V_PRINT]
975 1079 2 AND NOT .SIGNAME
976 1080 2 AND .SIGNAME[STSSV_FAC_NO] EQL SUBMIT$_FACILITY
977 1081 2 THEN
978 1082 2 SIGNAME[STSSV_FAC_NO] = PRINT$_FACILITY;
979 1083 2
980 1084 2
981 1085 2 ! Save the most severe condition as the exit status.
982 1086 2
983 1087 2 IF
984 1088 2 NOT .SIGNAME ! If an error signal
985 1089 4 AND ((.SIGNAME[STSSV_SEVERITY] ! and severity is worse
986 1090 4 GTRU .EXIT_STATUS[STSSV_SEVERITY])
987 1091 3 OR .EXIT_STATUS[STSSV_SEVERITY]) ! or no errors yet
988 1092 2 THEN
989 1093 2 EXIT_STATUS = .SIGNAME; ! then save it for exit
990 1094 2
991 1095 2
```



SUBMIT  
V04-000

PRINT and SUBMIT Commands

C 13  
16-Sep-1984 00:28:51  
14-Sep-1984 12:09:55

VAX-11 Bliss-32 V4.0-742  
[CLIUTL.SRC]SUBMIT.B32;1

Page 33  
(6)

: 992  
: 993  
1096 2 SSS\_RESIGNAL  
1097 1 END;

! Resignal to get message

00000000G	8F	02	A0	50	04	52	0000'	CF	9E	00002	HANDLER: .WORD	Save R2	: 1039
						AC		04	C1	00007	MOVAB	EXIT_STATUS, R2	: 1072
						19	04	A2	E9	0000C	ADDL3	#4, SIG, R0	: 1078
						2B		60	E8	00010	BLBC	FLAGS, 1\$	: 1079
						0C		00	ED	00013	BLBS	(R0), 3\$	: 1080
								0A	12	0001D	CMPZV	#0, #12, 2(R0), #SUBMITS_FACILITY	: 1082
02	A0		0C			00	00000000G	8F	F0	0001F	BNEQ	1\$	: 1088
						12		60	E8	00029	INSV	#PRINTS_FACILITY, #0, #12, 2(R0)	: 1090
	51		62			03		00	EF	0002C	BLBS	(R0), 3\$	: 1091
	51		60			03		00	ED	00031	EXTZV	#0, #3, EXIT_STATUS, R1	: 1093
								03	1A	00036	CMPZV	#0, #3, (R0), R1	: 1097
						03		62	E9	00038	BGTRU	2\$	: 1099
						62		60	D0	0003B	BLBC	EXIT_STATUS, 3\$	: 1099
						50	0918	8F	3C	0003E	MOVL	(R0), EXIT_STATUS	: 1099
								04	00043		MOVZWL	#2328, R0	: 1099
											RET		: 1099

; Routine Size: 68 bytes, Routine Base: CODE + 0975

SUBMIT  
V04-000

PRINT and SUBMIT Commands

D 13  
16-Sep-1984 00:28:51  
14-Sep-1984 12:09:55

VAX-11 Bliss-32 V4.0-742  
[CLIUTL.SRC]SUBMIT.B32;1

Page 34  
(7)

: 995  
: 996  
1098 1 END  
1099 0 ELUDOM

.EXTRN LIB\$SIGNAL, LIB\$STOP

PSECT SUMMARY

Name	Bytes	Attributes
DATA	12	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
CODE	2489	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	107	1	581	00:00.9

COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:SUBMIT/OBJ=OBJ\$:SUBMIT MSRC\$:SUBMIT/UPDATE=(ENH\$:SUBMIT)

: Size: 1625 code + 876 data bytes  
: Run Time: 00:32.4  
: Elapsed Time: 01:47.3  
: Lines/CPU Min: 2037  
: Lexemes/CPU-Min: 19676  
: Memory Used: 440 pages  
: Compilation Complete



0059

AH-BT13A-SE  
VAX/VMS V4.

**DIGITAL  
CONFIDE**

EQUIPMENT  
TIAL AND

CORPORATION  
PROPRIETARY